Enhanced Parking Space Monitoring System

ECE 445 Design Document 2

Team #41
TA: Chi Zhang
April 17, 2020

Patrick Connelly (prc2) - Wifi Chip and Neural Network Programming
Ben Wasicki (wasicki2) - Cloud Server and Website Development
John Scholl (johnts2) - Circuit Design and Website Development

Table of Contents

1. Introduction **2**
- 1.1 Objective 2
- 1.2 Background 2
- 1.3 Visual Aid 3
- 1.4 High Level Requirements 3

2. Design **4**
- 2.1 Block Diagram 4
- 2.2 Physical Design 5
- 2.3 Subsystems 5
  - 2.3.1 Camera 5
  - 2.3.2 Raspberry Pi 6
  - 2.3.3 LED Array 7
  - 2.3.4 Wifi Chip 8
  - 2.3.5 Power Module 9
  - 2.3.6 Cloud Server 9
  - 2.3.7 Website 10
- 2.4 Tolerance Analysis 11
  - 2.4.1 Circuit Schematic 11
  - 2.4.2 Component Tolerances 11

3. Cost and Schedule **12**
- 3.1 Cost Analysis 12
- 3.2 Schedule 13

4. Discussion of Ethics and Safety **14**

References **15**

1. Introduction

1.1 Objective

There are many times when it can be difficult for people to find parking, especially in cities or parking garages. It can be extremely frustrating to circle a parking garage multiple times without finding a spot for several minutes. Extra time spent searching for a spot creates unnecessary traffic and wastes fuel. In addition, drivers searching for open parking spaces often are not paying full attention to the road ahead of them. This creates a hazardous environment for both drivers and nearby pedestrians. Due to this, a product that aids drivers in locating a parking spot would be beneficial.

Our proposed solution is to monitor the parking spaces within a parking garage with a camera, and provide an easy to use web interface that would show where available parking spots are located. This way, a person could easily go to the website, find where a parking spot is located, and immediately go there, cutting the search time down drastically. In addition, colored LEDs will indicate the status of a spot to nearby drivers to further reduce their need to search for open spots. Mounting the device on the ceiling of the parking garage as seen in *Figure 1* allows for a better angle of sight and provides a degree of protection from vandalism or theft.

1.2 Background

According to INRIX, a parking and driving data analytics company, Americans spend an average of 17 hours per year searching for parking. This extra time costs about $345 per driver in wasted time and fuel. INRIX also found that 40% of drivers report avoiding going to shops due to the hassle of finding parking [5]. Burdensome parking experiences not only annoy drivers but also hurt the economy and local businesses as a whole. As well as being economically disruptive, poorly operated parking lots can also be dangerous. The National Safety Council reports that there are more than 50,000 car accidents per year in parking lots and garages. These accidents result in an average of 60,000 injuries and 500 deaths every year. With our design drivers will be able to be less distracted while getting to a parking spot.

Currently implemented solutions to this problem do not provide the same level of accuracy or convenience as our design. Occasionally one can find signs outside of a parking garage that indicates the number of available parking spaces. These counters are prone to inaccuracy when a garage is busy and cars are tightly packed [7]. Newer parking garages will display colored LEDs above each spot to show availability, but lack the accuracy and online display of our design.

When compared to the previous group's project, our solution differs in several key areas. Instead of using an IR sensor, our design utilizes a camera and video processing software to detect the presence of a vehicle. The previous solution required the bumper of the vehicle to be unrealistically close to the sensor in order to be counted. Our design would allow for more accurate detection and more variation in vehicle sizes. The previous design could also send a false positive in the case of a pedestrian standing in front of the IR sensor. The video processing approach of our design would eliminate such errors. In addition, we would provide a web interface so that a person could view the availability of various spots. This would function as a website where one could view available spots on a map, and easily see where the nearest available spot is.

1.3 Visual Aid

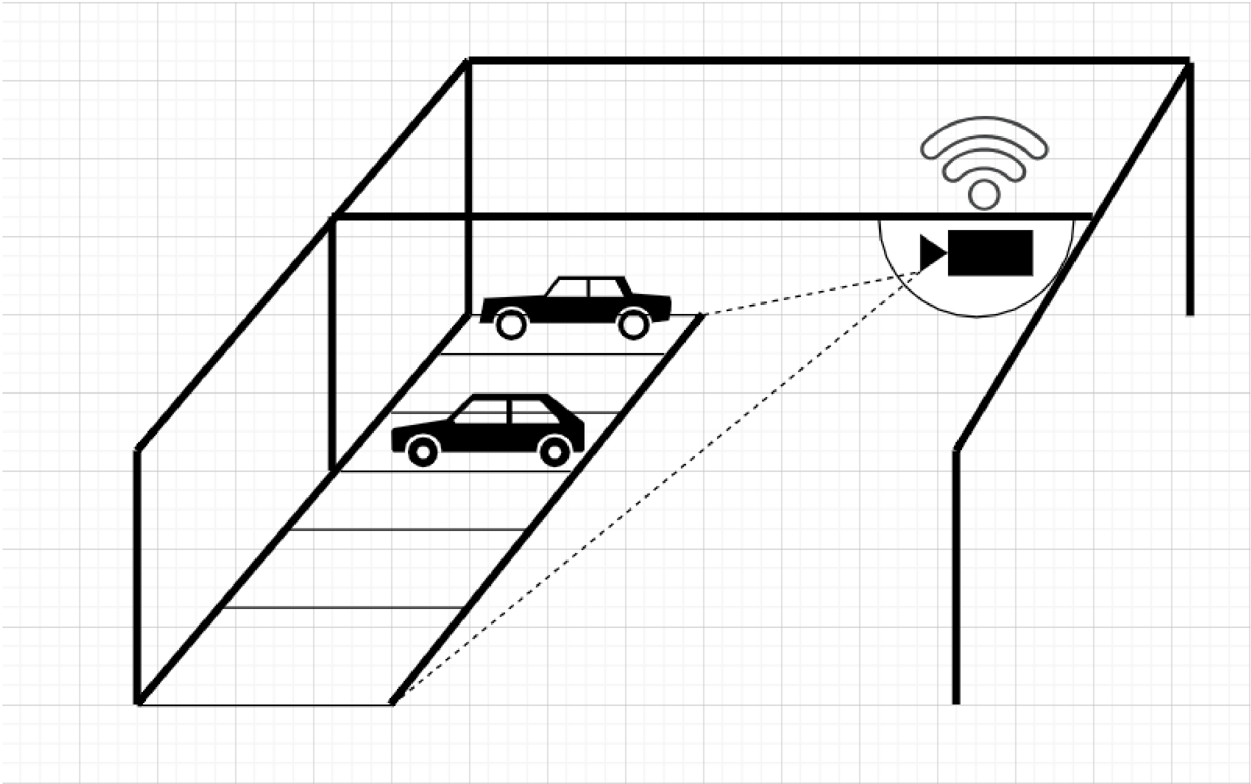

Figure 1. Enhanced Parking Space Monitoring System Visual Aid

1.4 High Level Requirements

The following are the most important qualities our project must exhibit in order to be successful:

- Be able to differentiate a vehicle from other objects.
- Detect the presence of a vehicle in a parking space with 90% accuracy.
- Conveniently notify users of available parking spots through LEDs near the spots and a website containing an up-to-date map.

2. Design

2.1 Block Diagram

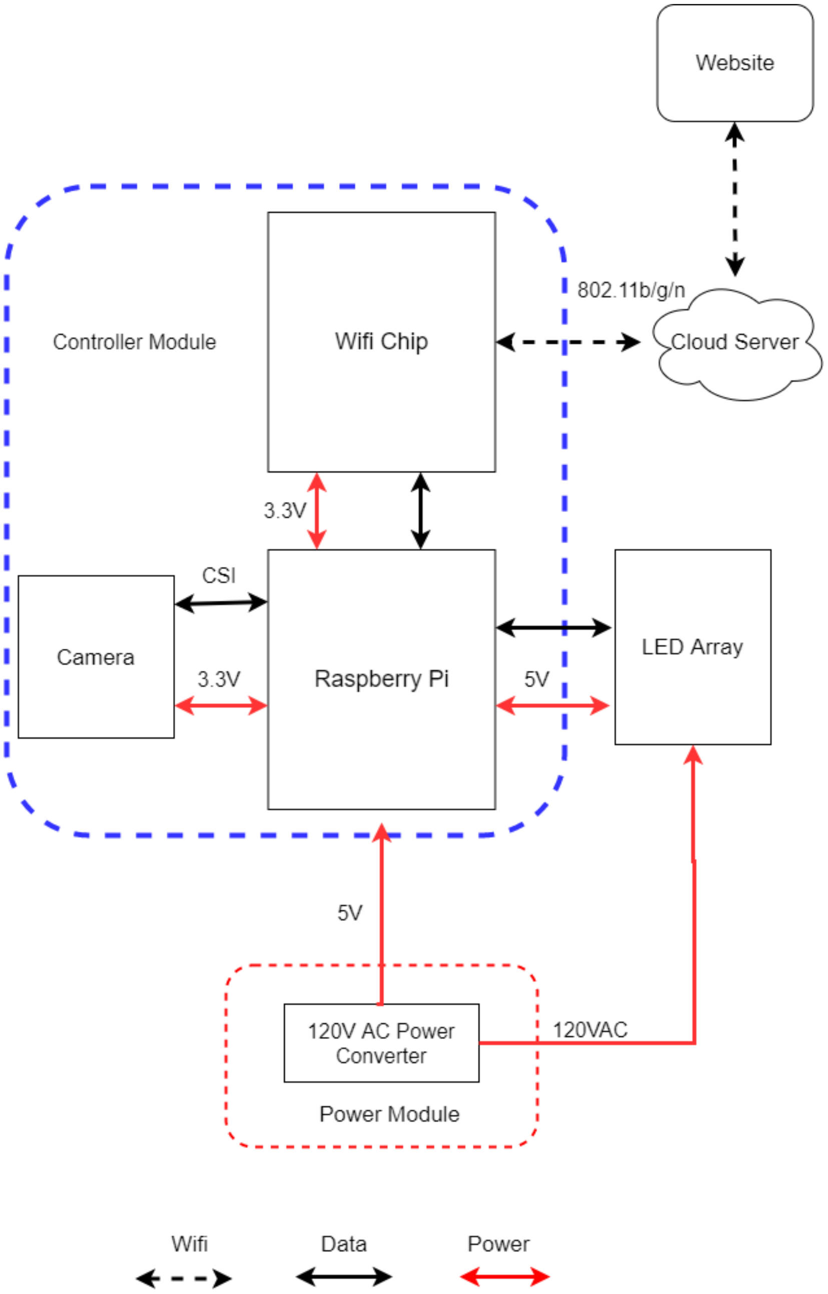

Figure 2. Enhanced Parking Space Monitoring System Block Diagram

2.2 Physical Design

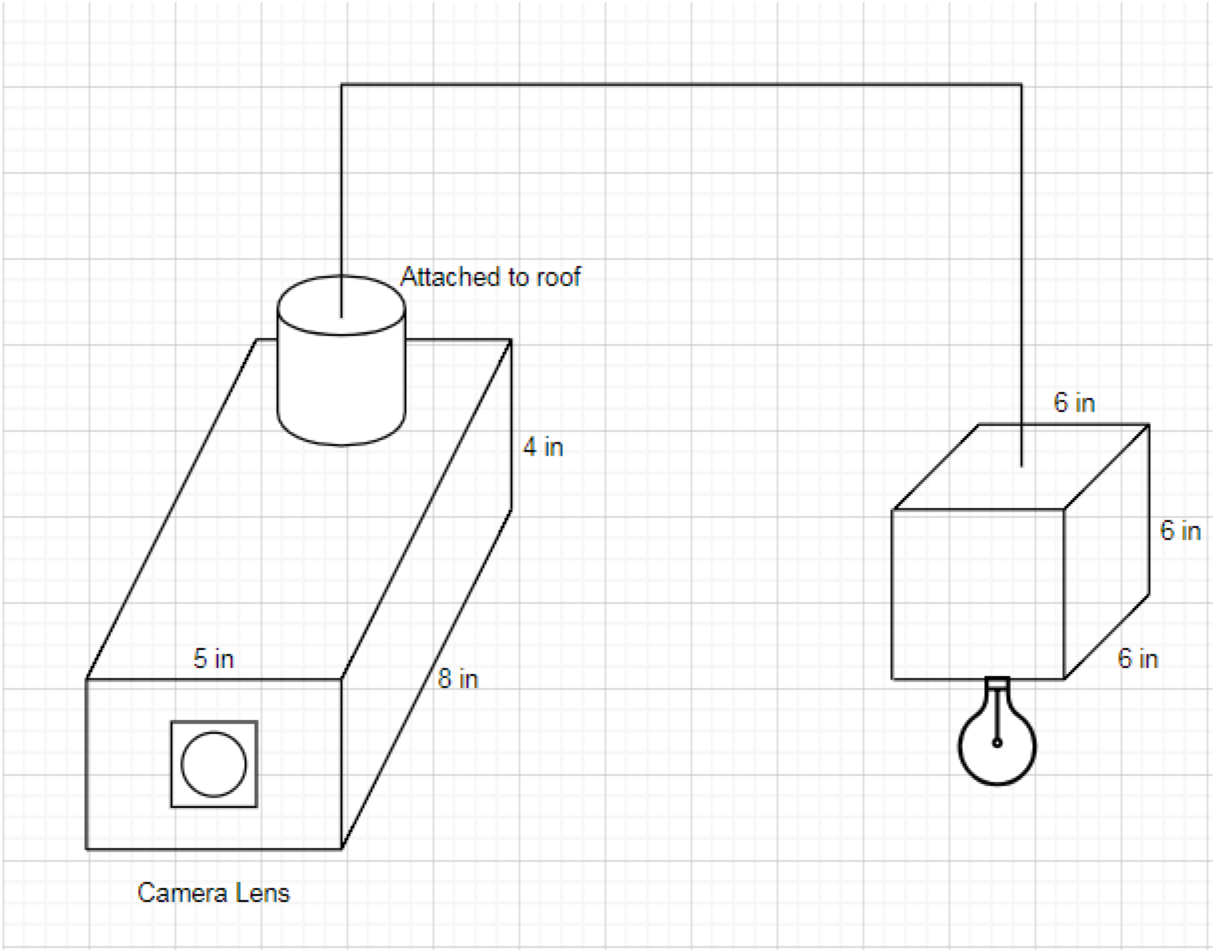

Fig. 3.Enhanced Parking Space Monitoring System Physical Design. Measurements are in inches

2.3 Subsystems

2.3.1 Camera

The Camera contributes primarily to the second high level requirement, since it is responsible for generating the images that allow software detection of vehicles to occur. We will use the Raspberry Pi Camera Module V2-8, which connects directly to the CSI port on the Raspberry Pi. It is 8-megapixels and capable of 1080p video and still images.The Camera will be responsible for sensing a parking area and sending a stream of image data to the Raspberry Pi every second. In order for the system to function properly, all parking spots will be visible by at least one camera.

Requirements	**Verification**
1. Send an image to the Raspberry Pi	1. A. Connect camera to Raspberry Pi B. Run software to take a picture and view it C. Verify the image displayed is as expected

2.3.2 Raspberry Pi

The Raspberry Pi contributes to the first and second high level requirements. It will be implemented with a Raspberry Pi 4 Model B 4GB. Not only is it responsible for deciding what is and is not a vehicle in an image, it must detect whether or not that vehicle is in a parking spot and report the status of every spot it is responsible for to the Wifi Chip. The data pins on the Raspberry Pi output at 3.3V, which is the necessary input for the Wifi Chip, so they can be directly connected. The Raspberry Pi will receive a constant stream of data from the camera module. It will analyze the images it receives using a pre-trained neural network, identifying all vehicles in the image and deciding whether or not they fall within the bounds of a parking spot.

For image processing, we will be forking a github project [8]. This project currently allows a user to specify the number of parking spots viewable by a camera feed, then subtracts the number of cars from the number of spots to get the available parking space number. Our fork would allow a user to specify what area of the image is used for each spot, and then only count a spot as in use if the car overlaps the defined parking space.

Requirements	**Verification**
1. Correctly identify the parking spots assigned to this Raspberry Pi 2. Correctly decide whether or not a car is in each of its parking spots 90% of the time 3. Process at least one frame every second at least 99% of the time 4. Transfer the status of the parking spots to the Wifi Chip exactly once every 10 seconds +/- 0.01 seconds	1. A. Load program to Raspberry Pi and connect to PC B. Get camera output such that all parking spots are empty C. Output number of empty spots to console and compare with actual count 2. A. Place a car in a parking spot for 100 seconds, then remove B. Read output of program over first 100 seconds. Verify it reads the spot as occupied for >= 90 seconds C. Read output of program for next 100

	seconds. Verify it reads the spot as unoccupied for >= 90 seconds D. Repeat for every spot assigned to camera E. Verify success for these tests in >= 90% of spots 3. A. Program requests current image from camera once a second B. Verify that program finishes before next request is made C. Repeat 99 more times, and verify all runs met this requirement 4. A. Use function to track time passed, and have variable that tracks current average parking lot status B. After 10 seconds, use parking status variable to set data line going to Wifi Chip

2.3.3 LED Array

The LED Array contributes to the third high level requirement, as it makes identifying open spots when at the parking area much easier. Every parking spot will have a red and green LED Array. Each array will connect back to its local Raspberry Pi to determine which array is active for that spot. The red array will inform the user the spot is taken, while the green array signals the spot as available. This will be controlled by a signal from the Raspberry Pi.

Requirements	**Verification**
1. For a red LED Array, only turn on when the associated parking spot is unavailable 2. For a green LED Array, only turn on when the associated parking spot is available	1. A. Hook 120V to the input of the LED array and gnd to ground. B. Send a 0V signal through the data input. C. Check to see if the red light is on and the green light is off. 2. A. Hook 120V to the input of the LED

	array and gnd to ground. B. Send a 5V signal through the data input. C. Check to see if the red light is off and the green light is on.

2.3.4 Wifi Chip

We will use the ESP8266 ESP-12E NodeMCU development module to handle all input processing and communication with the server. This will take data input from the Raspberry Pi and communicate the parking spot occupancy to the server. This module requires 3.3V to function, but there is an onboard regulator that tolerates up to 10V. Thus, we can safely use the 5V output from the Power Module. While transmitting, the ESP8266 WiFi chip uses between 120mA and 170mA, depending on transmission power. Receiving will use between 50mA and 56mA. Thus, there is relatively minimal power draw compared to the other components.

Requirements	**Verification**
1. Must be able to communicate with Cloud Server over IEEE 802.11b/g/n at a 1 Kbps data rate 2. Must be able to take input from Raspberry Pi 3. Must send and receive a packet upon pushing the test button	1. A. Flash the network credentials and a test program to the board through the USB port with the “Flash” button B. Press the test button. The Wifi chip will send 1 Mb of data to the server, which will time the transfer. C. Check the server test logs and ensure the transfer time was under 10 seconds. 2. A. Flash test program to board such that onboard LED is turned on by data pin 3. A. Set up a listener program on the same network as the Wifi Chip B. Flash the network credentials and a test program pointing to the listener to the board through the USB port with the “Flash” button C. Press the test button. The Wifi chip will send a packet to the listener D. Check that the onboard test LED is turned on

2.3.5 Power Module

The Power Module contributes to all three high level components because it is responsible for powering the hardware components by converting 120V AC to a form usable by the Camera, Raspberry Pi, and Wifi Chip. The Power Module is responsible for converting the 120V AC output from a standard wall socket to a format usable by the camera, Raspberry Pi, and Wifi Chip.

Requirements	**Verification**
1. Accept 120V AC from a wall socket and output 5V AC.	1. A. Connect a voltage probe to the output and ground. B. Check to see if the output is within 5% (+/- 0.25V) of 5VAC.

2.3.6 Cloud Server

The Cloud Server contributes to the third high level requirement. It collects and sorts data gathered by the hardware devices. The Cloud Server will accumulate the processed image data from the hardware Raspberry Pi. From there it will update the website with up-to-date information on where available parking spaces are located.

Requirements	**Verification**
1. The server will be able to receive data from at least one hardware component. 2. The server will be able to take data that is collected from at least one hardware component and update the website with the latest parking information.	1. A. Launch *server* on google cloud. B. Launch *servertest* program with arg: *"test1"* which will attempt to send data to the server via TCP. C. If the program successfully sends the data without a TCP timeout, it will return *"success!"* because the server could be reached. 2. A. Launch *server* on google cloud. B. Launch *website* program on google cloud. C. Make sure that the website requirement 2 is working. D. Launch *servertest* program with arg:

	“test2” which will attempt to send data to the server every 10 seconds such that the parking map will fill in one at a time. E. Navigate to the website in a browser and refresh to see if the parking map is getting updated every 10 seconds.

2.3.7 Website

The Website contributes to the third high level requirement. It takes parking space data from the Cloud Server and displays it in a readable format to consumers. The Website will host a map showing the latest information on which parking spaces are available. It will receive this data from the Cloud Server.

Requirements	**Verification**
1. The website will be able to receive data from the server component. 2. The website will be able to take data received from the server and display it in a readable map format for consumers.	1. A. Launch *website* program on google cloud. B. Launch *webtest* program with arg: *“test1”* which will attempt to send data to the website via TCP. C. If the program successfully sends the data without a TCP timeout, it will return *“success!”* because the website could be reached. 2. A. Launch *website* program on google cloud. B. Launch *webtest* program with arg: *“test2”* which will attempt to send data to the website every 10 seconds such that the parking map will fill in one at a time. C. Navigate to the website in a browser and refresh to see if the parking map is getting updated every 10 seconds.

2.4 Tolerance Analysis

2.4.1 Circuit Schematic

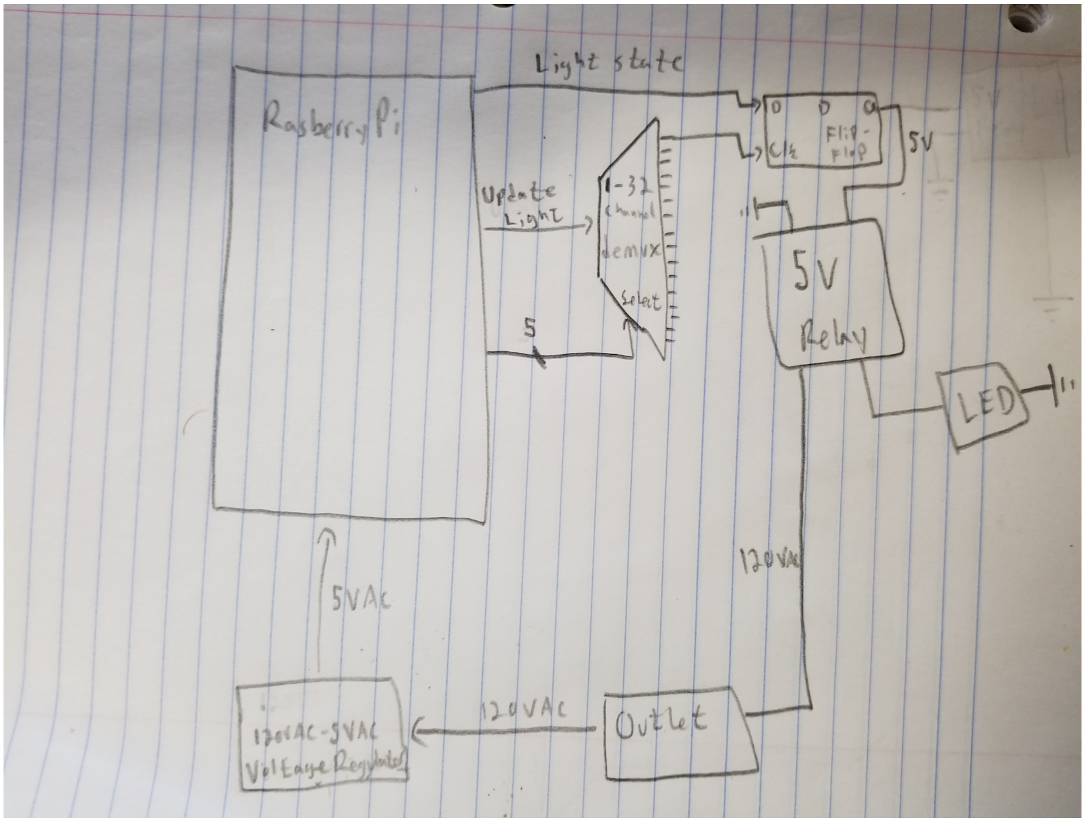

Fig. 5. Enhanced Parking Space Monitoring System Circuit Schematic

2.4.2 Component Tolerances

The Raspberry Pi Model B has a bare-board draw of 500mA and a recommended PSU current capacity of 1.2A. It's operable voltage ranges are 4.75-5.25V.

3. Cost and Schedule

3.1 Cost Analysis

We estimated the cost of our development assuming a 10 hour per week of a 16 week schedule. Considering the average starting salary of a BE in ECE, we calculated our hourly wage to be $45 per hour [4].

$$\frac{\$45}{hr} \cdot \frac{10hr}{wk} \cdot 16wk \cdot 3 \ \cdot 2.5 \ = \$54,000$$

Description	Part Number	Manufacturer	Module	Price
Raspberry Pi 4 Model B (4GB)	B07TC2BK1X	Raspberry Pi	Control	$61.70
Raspberry Pi Camera Module V2-8	B01ER2SKFS	Raspberry Pi	Control	$27.50
Wifi Chip	ESP8266 ESP-12E NodeMCU	MakerFocus	Control	$9.39
Green LED x4	A19 Christmas Led Light Bulbs	NOVELUX	LED Array	$25.94
Voltage Regulator	296-20778-2-ND	Texas Instruments	Power	$2.39
E26 Light Bulb Socket x4	99770033	BROAN	LED Array	$23.60

Total Part Cost: $155.48

Total Development Cost: $54,155.48

3.2 Schedule

Week	Ben	Patrick	John
1	Begin website development	Work on Parking Spot Program	Begin website development
2	Get website to display parking data in a text-based format	Work on Parking Spot Program	Test parts when they arrive to assure operation within requirements
3	Website Debug week	Work on Parking Spot Program	Get Power Module operating as required and link LED array to Raspberry Pi
4	Begin server development.	Test program on Raspberry Pi with preloaded video	Debug Power module and LED array
5	Make sure server can send data to website.	Hook up Camera to Raspberry Pi and switch to live feed processing	Debug circuit design
6	Make sure server can receive data from hardware.	Hook up Raspberry Pi to Wifi Chip	Start making website more user friendly(css/add map).
7	Server debug week	Connect Wifi Chip to server	Continue website development.
8	Help make website more user friendly(css/add map)	Ensure end-to-end functionality with camera and website	Continue website development.
9	Finish website development and begin final testing and debugging testing	Hardware and Wifi debugging	Finish website development and begin final testing and debugging testing
10	Finish final testing and debugging testing	Finish final testing and debugging testing	Finish final testing and debugging testing

4. Discussion of Ethics and Safety

As the developers of this project, we believe it is important that we produce a safe, reliable, and efficient product to our user. We commit ourselves to holding a high degree of professional conduct in accordance with both the IEEE and ACM Code of Ethics. We will avoid ethical breaches by following all device specifications, working in our respective areas of competence, and clearly stating proper operating procedure (ACM 2.6). At the same time, we acknowledge that our device could be misused; therefore, we will take all necessary precautions to prevent any harmful modes of operation.

In accordance with the ACM Code of Ethics, this project will pose no risk to the user or community under standard operations. We will ensure that all wireless protocols are followed, and communications will be secure. The data gathered by our sensor will be the sole property of the intended user of the device (ACM 2.9). All software will follow accepted community standards.

Our module will have a camera attachment. Since a parking lot or parking garage is a public place where one does not have a reasonable expectation of privacy, cameras are allowed to record without any posted signage by law [3]. It may be nice, however, for consumers of the parking lot or garage to have a notice that cameras are active posted via sign.

It is both illegal and unsafe to use one’s phone while driving [4]. As such, on the sign that gives the URL to the parking map website, we will post notices to either pull off to the side and stop while accessing the website or have a passenger navigate the website.

In addition, we will ensure there is no exposed wiring or electrical components in our design to minimize the risk of electrical shock, especially due to the fact that our project uses 120V from a wall outlet. Similarly we will ensure all components are operating within their respective operating regions to reduce the risk of a short or fire hazard. We will also be mounting the hardware components with screws, mitigating any risk of the hardware falling and causing damage to vehicles or people in the area.

References

[1] “IEEE Code of Ethics,” *IEEE*. [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8.html. [Accessed: 2-Apr-2020].

[2] “ACM Code of Ethics and Professional Conduct,” *Association for Computing Machinery*. [Online]. Available: https://www.acm.org/code-of-ethics. [Accessed: 1-Apr-2020].

[3] “The Law of Video Surveillance,” *The Law Offices Paul Samakow.* [Online]. Available: https://www.samakowlaw.com/articles/law-video-surveillance [Accessed: 3-Apr-2020].

[4] “Distracted Driving Laws,” *drive safely.* [Online]. Available: https://www.idrivesafely.com/driving-resources/laws/distracted-driving [Accessed: 3-Apr-2020].

[5] K. McCoy, “Drivers spend an average of 17 hours a year searching for parking spots,” *CNBC*, 12-Jul-2017. [Online]. Available: https://www.cnbc.com/2017/07/12/drivers-spend-an-average-of-17-hours-a-year-searching-for-parking-spots.html. [Accessed: 02-Apr-2020].

[6] “Why hundreds are killed in crashes in parking lots and garages every year,” *CBS News*, 21-Nov-2016. [Online]. Available: https://www.cbsnews.com/news/parking-lot-accidents-distracted-drivers-national-safety-council/. [Accessed: 02-Apr-2020].

[7] D. Roos, “How Parking Garages Track Open Spaces, and Why They Often Get It Wrong,” *HowStuffWorks*, 14-Apr-2017. [Online]. Available: https://electronics.howstuffworks.com/everyday-tech/how-parking-garages-track-open-spaces-why-they-often-get-it-wrong.htm. [Accessed: 02-Apr-2020].

[8] Ankit Khare, “Smart-Park-with-YOLO-V3,” *Github*, 10-Feb-2019. [Online]. Available: https://github.com/ankit1khare/Smart-Park-with-YOLO-V3. [Accessed: 11-Apr-2020].

[9] "Raspberry Pi 4 Model B 2019 Quad Core 64 Bit WiFi Bluetooth (4GB)", *Amazon.com*, 2020. [Online]. Available: https://www.amazon.com/Raspberry-Model-2019-Quad-Bluetooth/dp/B07TC2BK1X?th=1. [Accessed: 16- Apr- 2020].

[10] "MakerFocus 2pcs ESP8266 NodeMCU", *Amazon.com*, 2020. [Online]. Available: https://www.amazon.com/Makerfocus-ESP8266-ESP-12E-Internet-Development/dp/B01IK9GEQG. [Accessed: 16- Apr- 2020].

[11] "Raspberry Pi Camera Module V2-8 Megapixel,1080p", *Amazon.com*, 2020. [Online]. Available: https://www.amazon.com/Raspberry-Pi-Camera-Module-Megapixel/dp/B01ER2SKFS. [Accessed: 16- Apr- 2020].

[12] "NOVELUX A19 Christmas Led Light Bulbs", *Amazon.com*, 2020. [Online]. Available: https://www.amazon.com/NOVELUX-Equivalent-Dimmable-Energy-Bulb-2PACK/dp/B07MS1KXTY/ref=sr_1_1?dchild=1&qid=1587184636&refinements=p_n_feature_twenty_browse-bin%3A3267897011&s=hi&sr=1-1. [Accessed: 16- Apr- 2020].

[13] "TL783CKTTR Texas Instruments", *Digikey.com*, 2020. [Online]. Available: https://www.digikey.com/product-detail/en/texas-instruments/TL783CKTTR/296-20778-1-ND/1204637. [Accessed: 16- Apr- 2020].

[14] "BROAN Light Socket", *Grainger.com*, 2020. [Online]. Available: https://www.grainger.com/product/BROAN-Light-Socket-25WU52. [Accessed: 16- Apr- 2020].

[15] "CAB-14092 SparkFun Electronics", *Digikey.com*, 2020. [Online]. Available: https://www.digikey.com/product-detail/en/sparkfun-electronics/CAB-14092/CAB-14092-ND/6833934. [Accessed: 17- Apr- 2020].